Case

- Time in My Pocket: Micro Time Management Application

- In today’s fast-paced world, time pressure has become a universal issue. Busy work schedules, long commutes, and constant digital distractions often leave people feeling like there aren’t enough hours in the day. “Time in My Pocket” offers an innovative solution by identifying unnoticed micro-time slots throughout the day and turning them into opportunities for productivity, learning, or relaxation. One of the app’s users, Mehmet Çelik, is a 35-year-old professional living in Istanbul who registered on February 26, 2025. Before using the app, Mehmet’s daily routine included hidden time sinks such as spending an hour aimlessly watching YouTube in the morning, late-night scrolling, and small idle moments like a 10-minute bus ride or 15 minutes stuck in traffic. After downloading “Time in My Pocket,” Mehmet learned to recognize and reclaim these moments, filling them with meaningful activities.

- The process begins with Mehmet entering his daily schedule—such as work hours, commute time, and evening downtime. With his permission, the app automatically detects time-wasting activities throughout the day. The built-in AI processes this data, calculating Mehmet's total free, busy, and wasted time. It then matches these insights with Mehmet's habits and preferences to generate personalized recommendations in real time. For instance, instead of watching random videos during his 10-minute morning bus ride, Mehmet now reads a focused article. While waiting seven minutes for lunch, he practices a quick breathing exercise. During a 15-minute traffic jam, he learns a new vocabulary word, and after dinner, he replaces endless scrolling with a short meditation session. If Mehmet skips or delays a suggestion, the AI adapts and offers alternatives the next day—for example, switching an article with a short podcast. It also adjusts to changes in time slot duration, ensuring each recommendation remains relevant and actionable. Mehmet can give brief feedback within the app, helping the system refine future suggestions. Admin users have the ability to monitor and adjust AI parameters if needed, while the AI engine constantly analyzes activity and updates suggestions in the background without disrupting the user experience. As a result, Mehmet no longer wastes his micro-time. He now reclaims an average of 30 minutes per day for personal growth, relaxation, and balance. "Time in My Pocket" stands out as a smart assistant that gives people back the time they didn't even know they had.

Microsoft®
SQL Server®

WHY?

-Advanced Temporal Data Handling

-Performance and Scalability

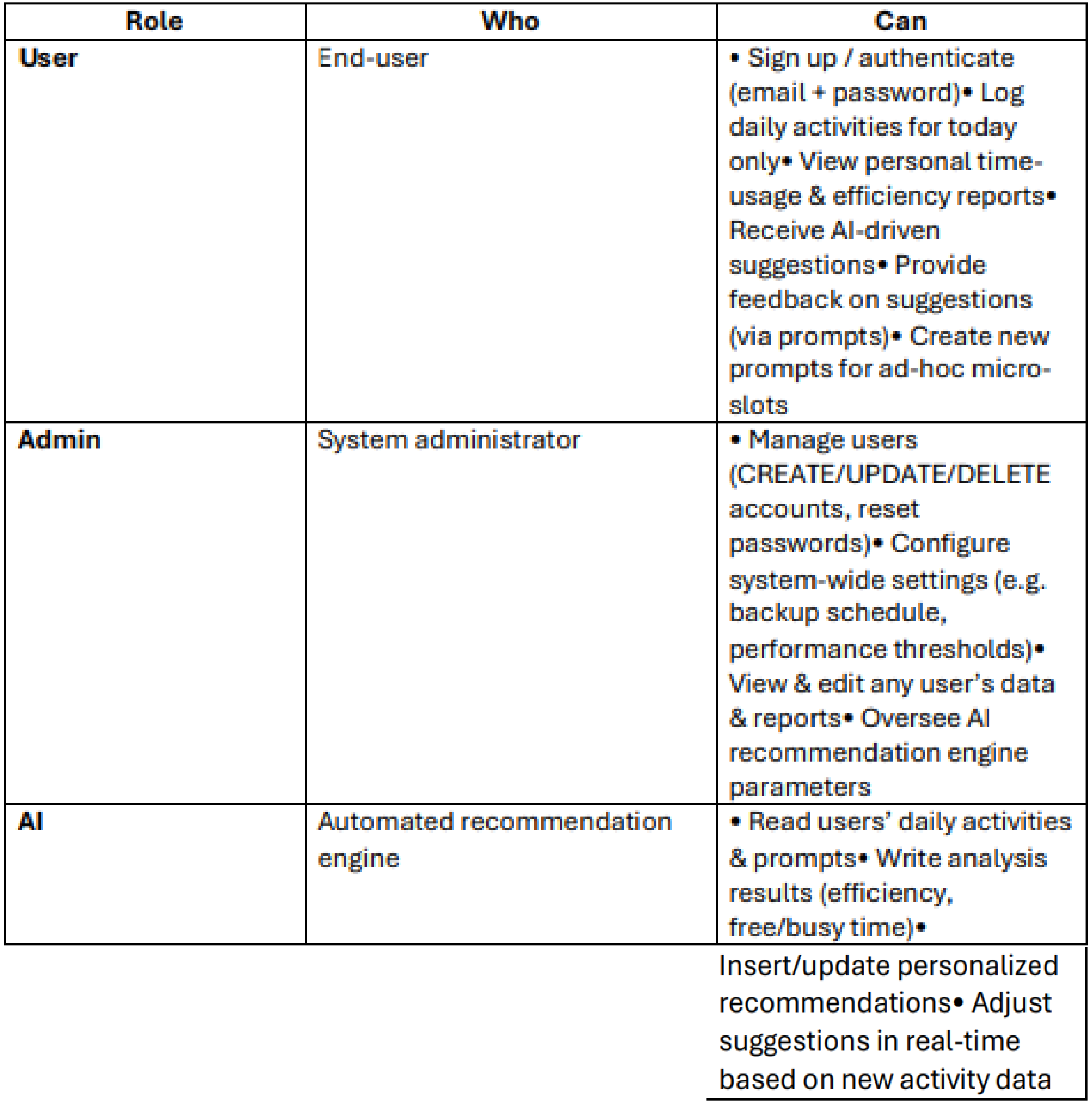
-Security and Governance

-JSON Support and Flexible Schema

-Seamless Integration with the Microsoft Ecosystem

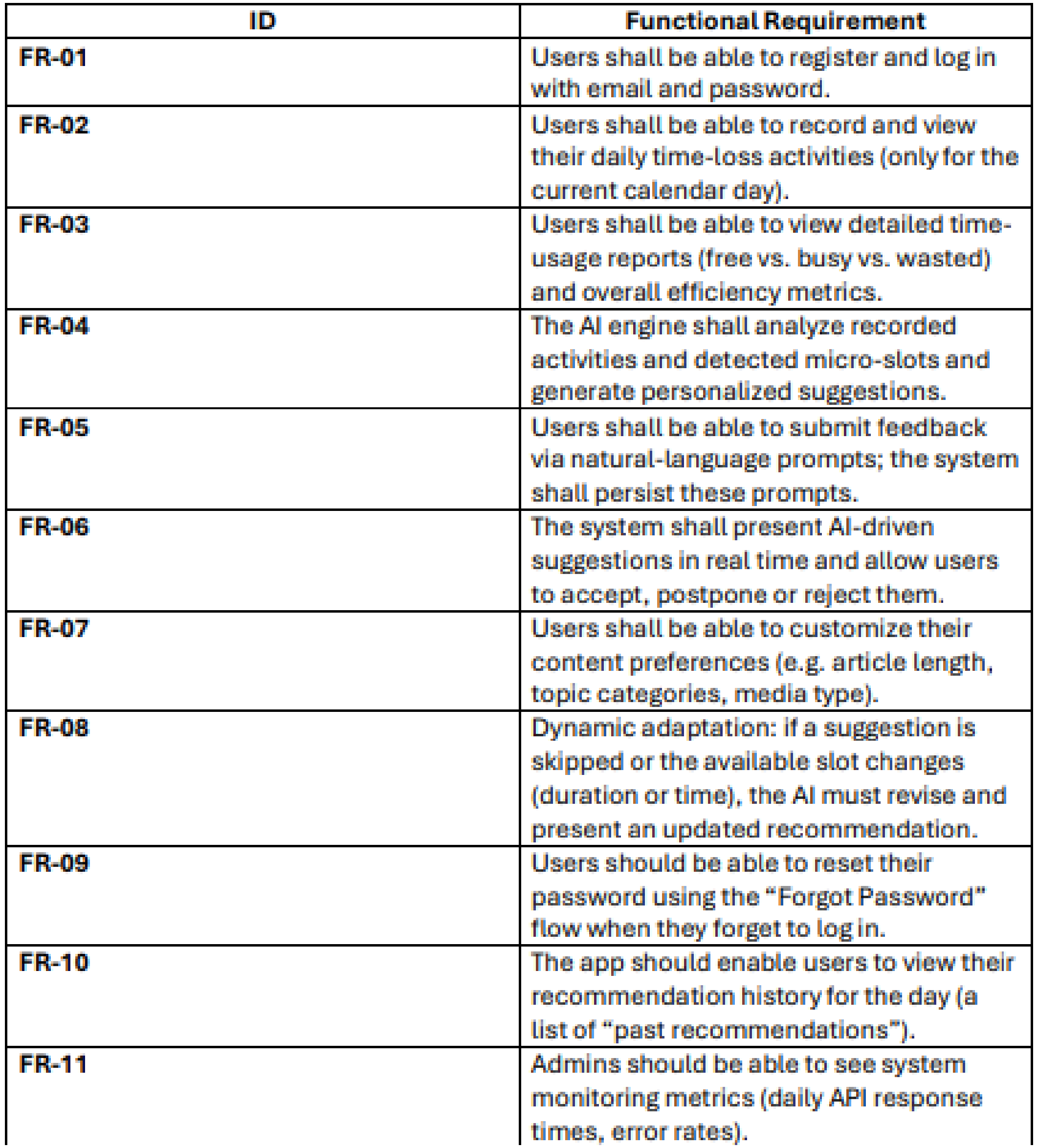
-Backup, High Availability, and Easy Maintenance

User Roles & Permissions



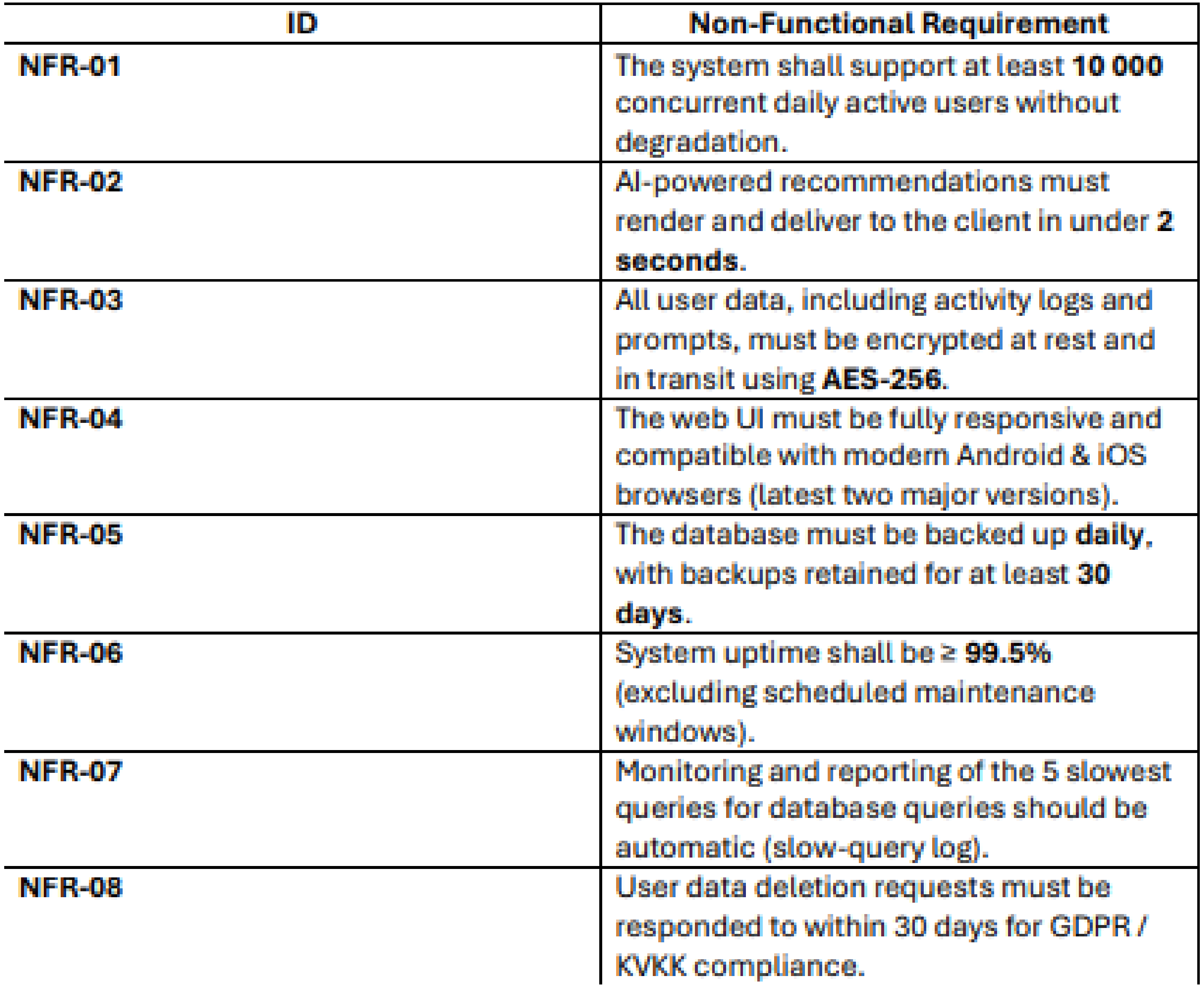
Role	Who	Can
User	End-user	• Sign up / authenticate (email + password)• Log daily activities for today only• View personal time-usage & efficiency reports• Receive AI-driven suggestions• Provide feedback on suggestions (via prompts)• Create new prompts for ad-hoc micro-slots
Admin	System administrator	• Manage users (CREATE/UPDATE/DELETE accounts, reset passwords)• Configure system-wide settings (e.g. backup schedule, performance thresholds)• View & edit any user's data & reports• Oversee AI recommendation engine parameters
AI	Automated recommendation engine	• Read users' daily activities & prompts• Write analysis results (efficiency, free/busy time)• Insert/update personalized recommendations• Adjust suggestions in real-time based on new activity data

Functional Requirements



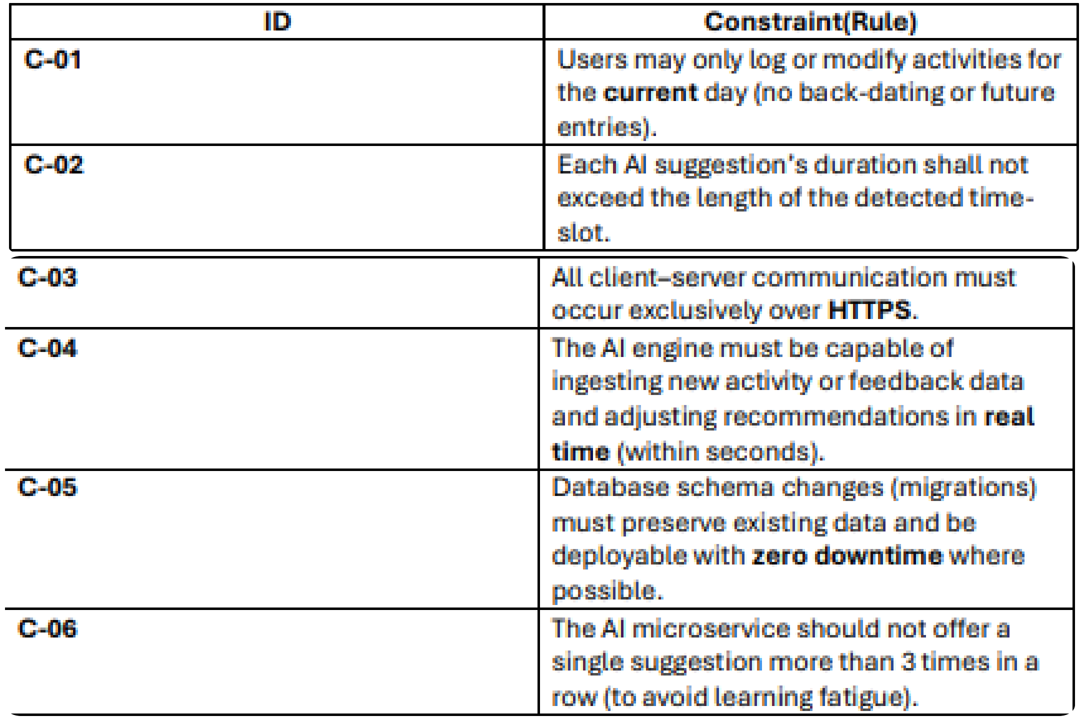
ID	Functional Requirement
FR-01	Users shall be able to register and log in with email and password.
FR-02	Users shall be able to record and view their daily time-loss activities (only for the current calendar day).
FR-03	Users shall be able to view detailed time-usage reports (free vs. busy vs. wasted) and overall efficiency metrics.
FR-04	The AI engine shall analyze recorded activities and detected micro-slots and generate personalized suggestions.
FR-05	Users shall be able to submit feedback via natural-language prompts; the system shall persist these prompts.
FR-06	The system shall present AI-driven suggestions in real time and allow users to accept, postpone or reject them.
FR-07	Users shall be able to customize their content preferences (e.g. article length, topic categories, media type).
FR-08	Dynamic adaptation: if a suggestion is skipped or the available slot changes (duration or time), the AI must revise and present an updated recommendation.
FR-09	Users should be able to reset their password using the “Forgot Password” flow when they forget to log in.
FR-10	The app should enable users to view their recommendation history for the day (a list of “past recommendations”).
FR-11	Admins should be able to see system monitoring metrics (daily API response times, error rates).

Non-Functional Requirements



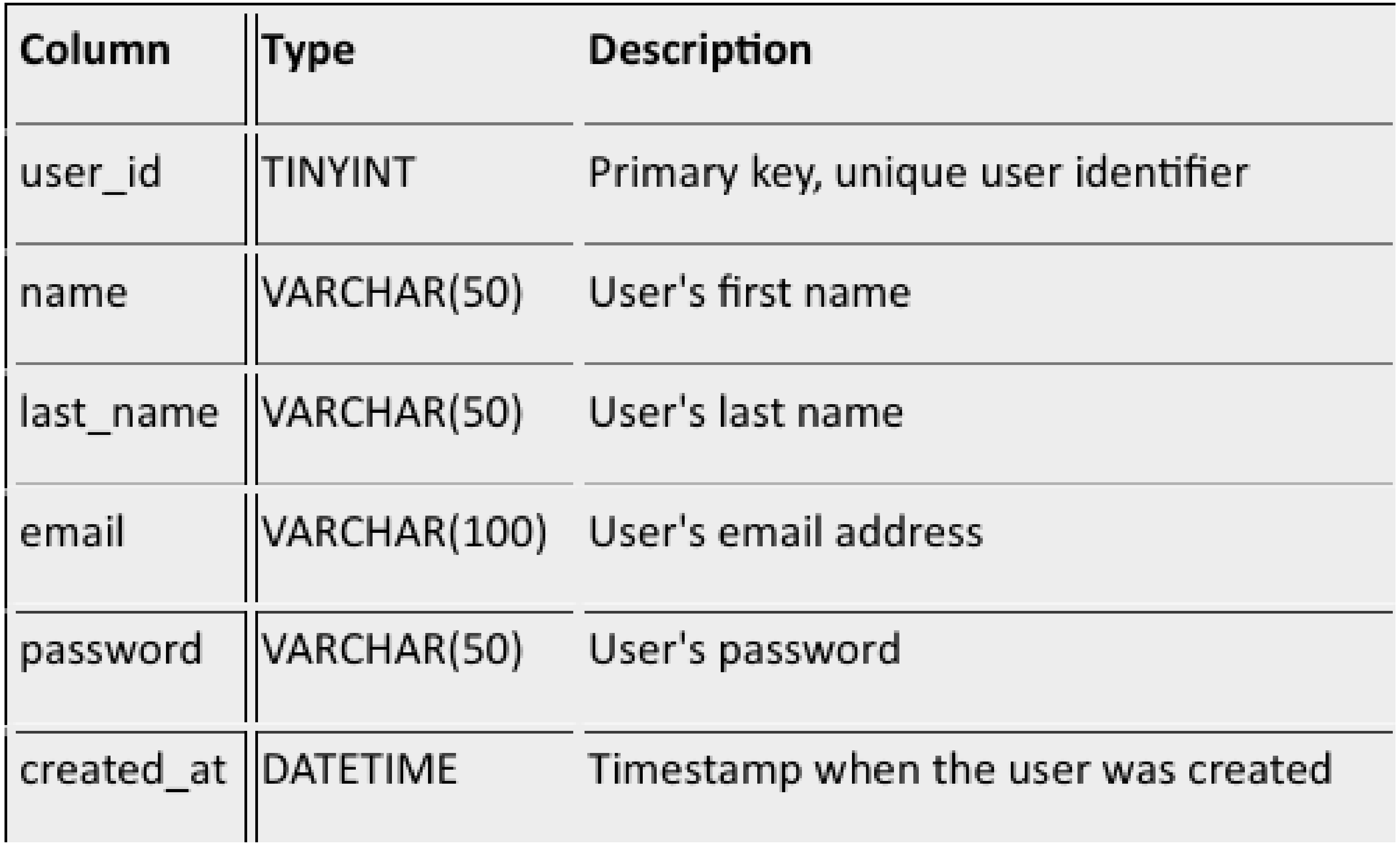
ID	Non-Functional Requirement
NFR-01	The system shall support at least **10 000** concurrent daily active users without degradation.
NFR-02	AI-powered recommendations must render and deliver to the client in under **2 seconds**.
NFR-03	All user data, including activity logs and prompts, must be encrypted at rest and in transit using **AES-256**.
NFR-04	The web UI must be fully responsive and compatible with modern Android & iOS browsers (latest two major versions).
NFR-05	The database must be backed up **daily**, with backups retained for at least **30 days**.
NFR-06	System uptime shall be ≥ **99.5%** (excluding scheduled maintenance windows).
NFR-07	Monitoring and reporting of the 5 slowest queries for database queries should be automatic (slow-query log).
NFR-08	User data deletion requests must be responded to within 30 days for GDPR / KVKK compliance.

Constraints



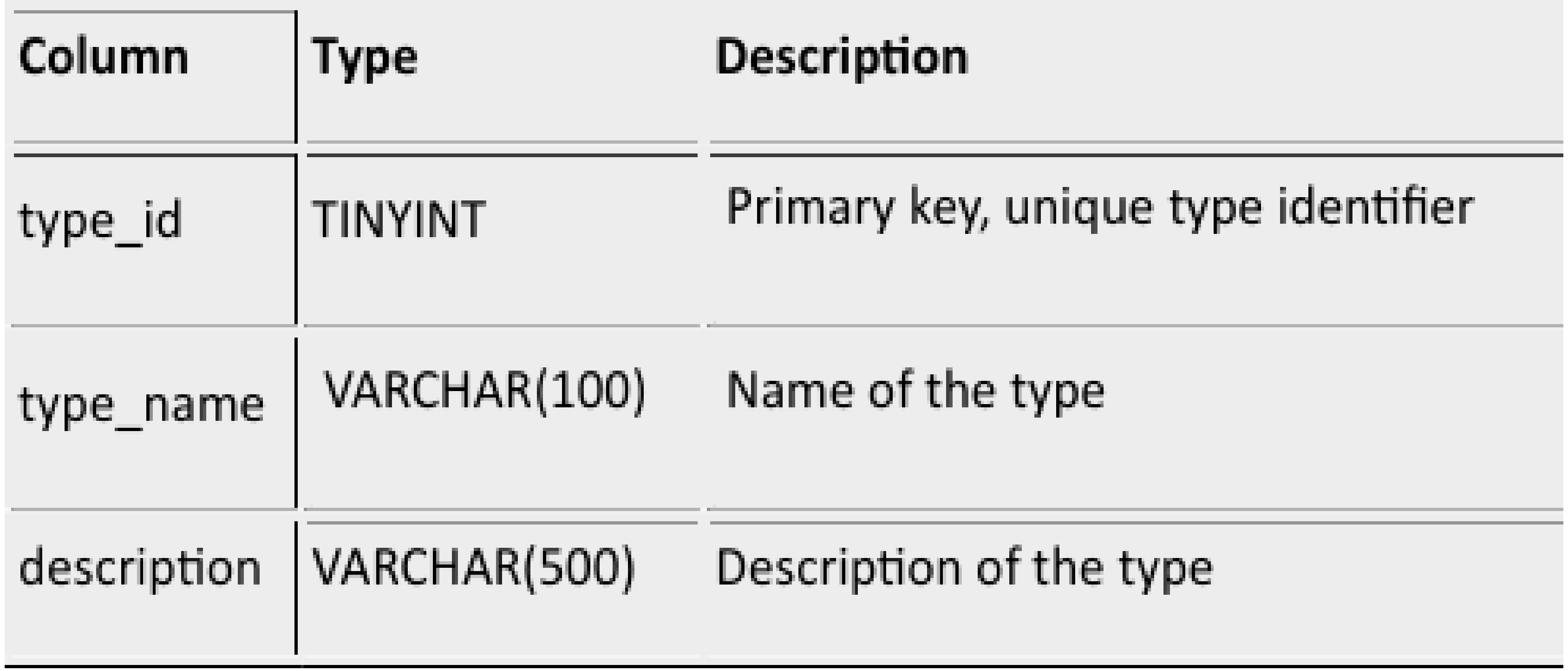
ID	Constraint(Rule)
C-01	Users may only log or modify activities for the **current** day (no back-dating or future entries).
C-02	Each AI suggestion's duration shall not exceed the length of the detected time-slot.
C-03	All client–server communication must occur exclusively over **HTTPS**.
C-04	The AI engine must be capable of ingesting new activity or feedback data and adjusting recommendations in **real time** (within seconds).
C-05	Database schema changes (migrations) must preserve existing data and be deployable with **zero downtime** where possible.
C-06	The AI microservice should not offer a single suggestion more than 3 times in a row (to avoid learning fatigue).

Users Table



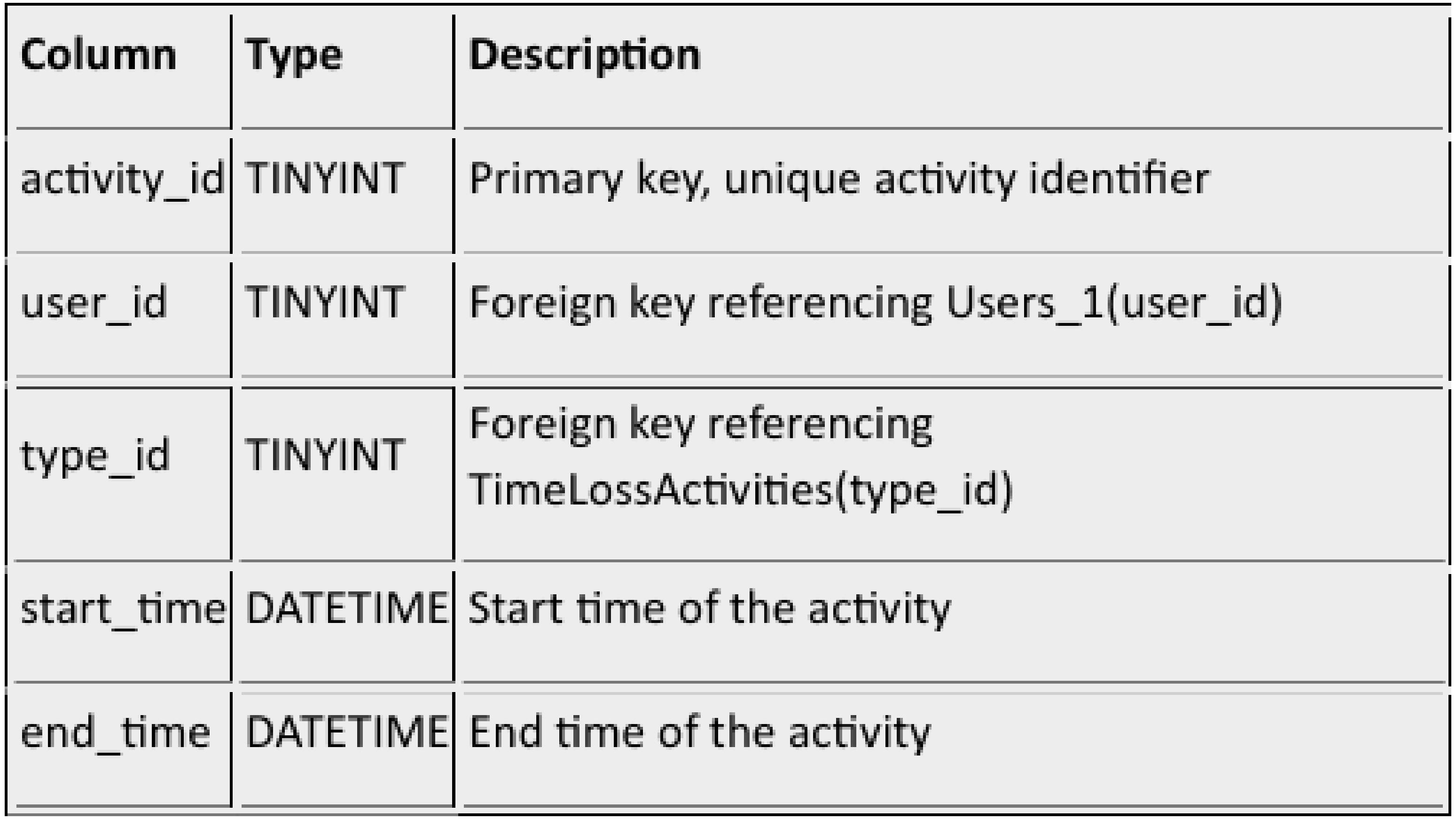
Column	Type	Description
user_id	TINYINT	Primary key, unique user identifier
name	VARCHAR(50)	User's first name
last_name	VARCHAR(50)	User's last name
email	VARCHAR(100)	User's email address
password	VARCHAR(50)	User's password
created_at	DATETIME	Timestamp when the user was created

Time Loss Activities Table



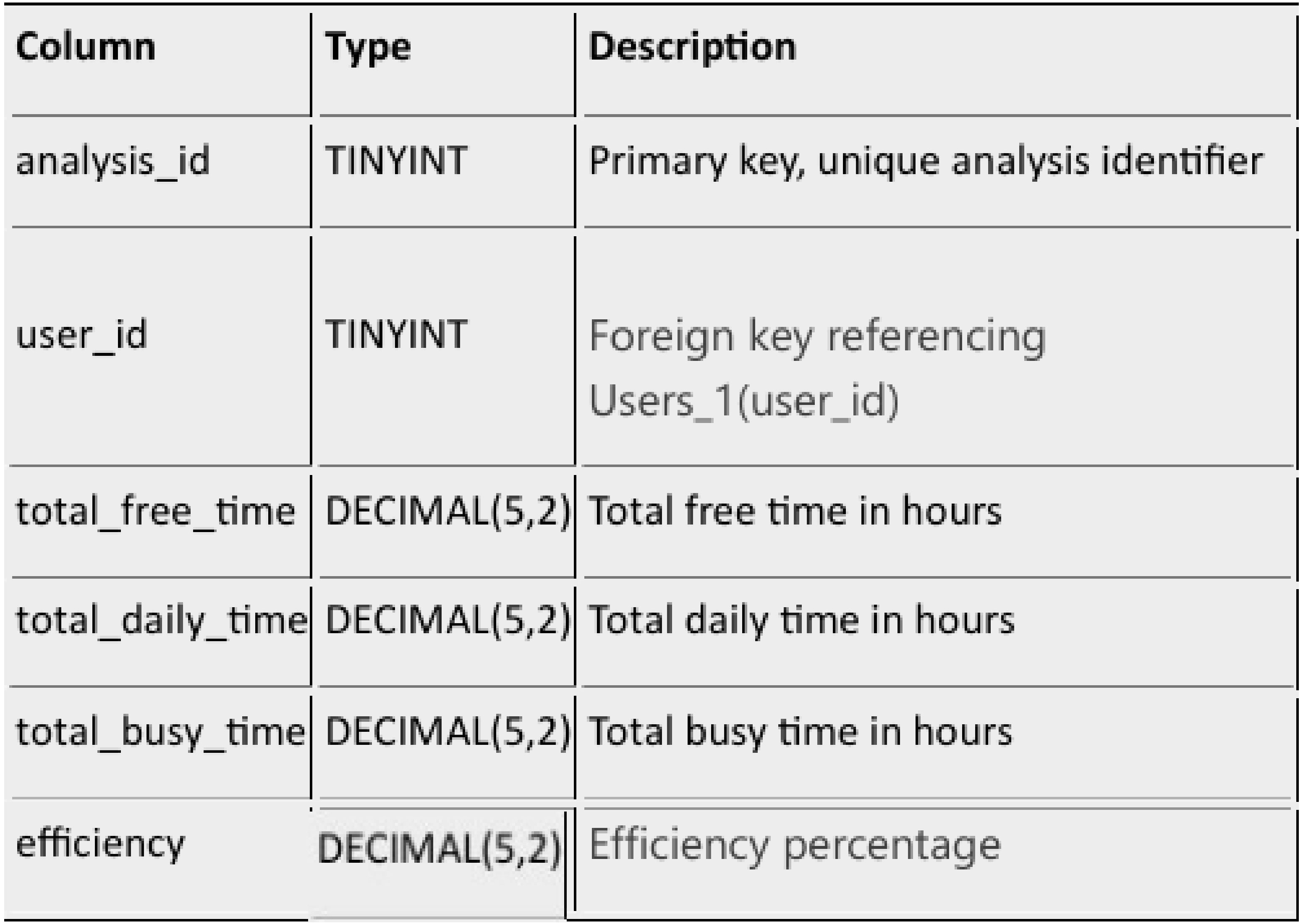
Column	Type	Description
type_id	TINYINT	Primary key, unique type identifier
type_name	VARCHAR(100)	Name of the type
description	VARCHAR(500)	Description of the type

Daily Time Loss Activities Table



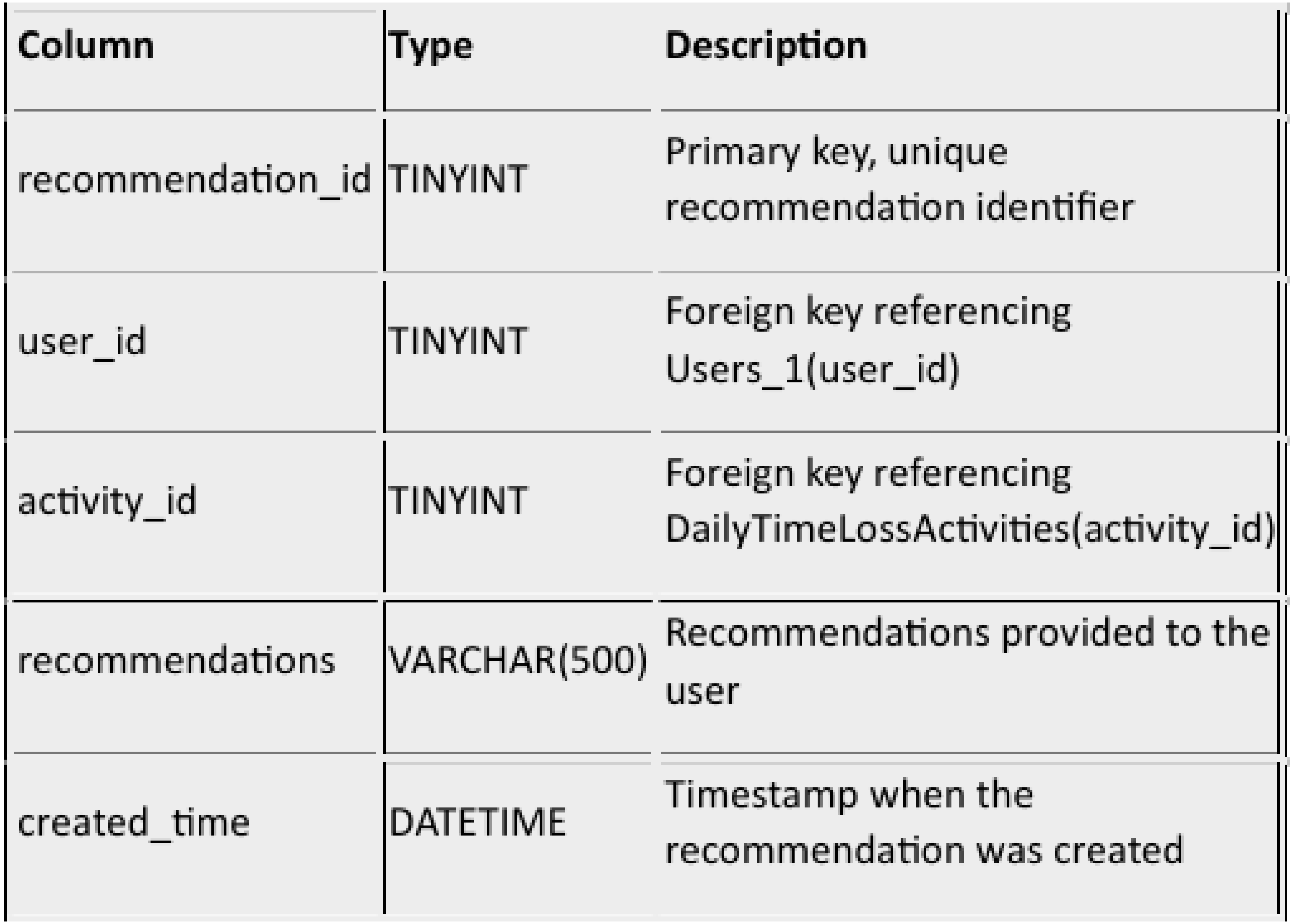
Column	Type	Description
activity_id	TINYINT	Primary key, unique activity identifier
user_id	TINYINT	Foreign key referencing Users_1(user_id)
type_id	TINYINT	Foreign key referencing TimeLossActivities(type_id)
start_time	DATETIME	Start time of the activity
end_time	DATETIME	End time of the activity

Analysis Results Table



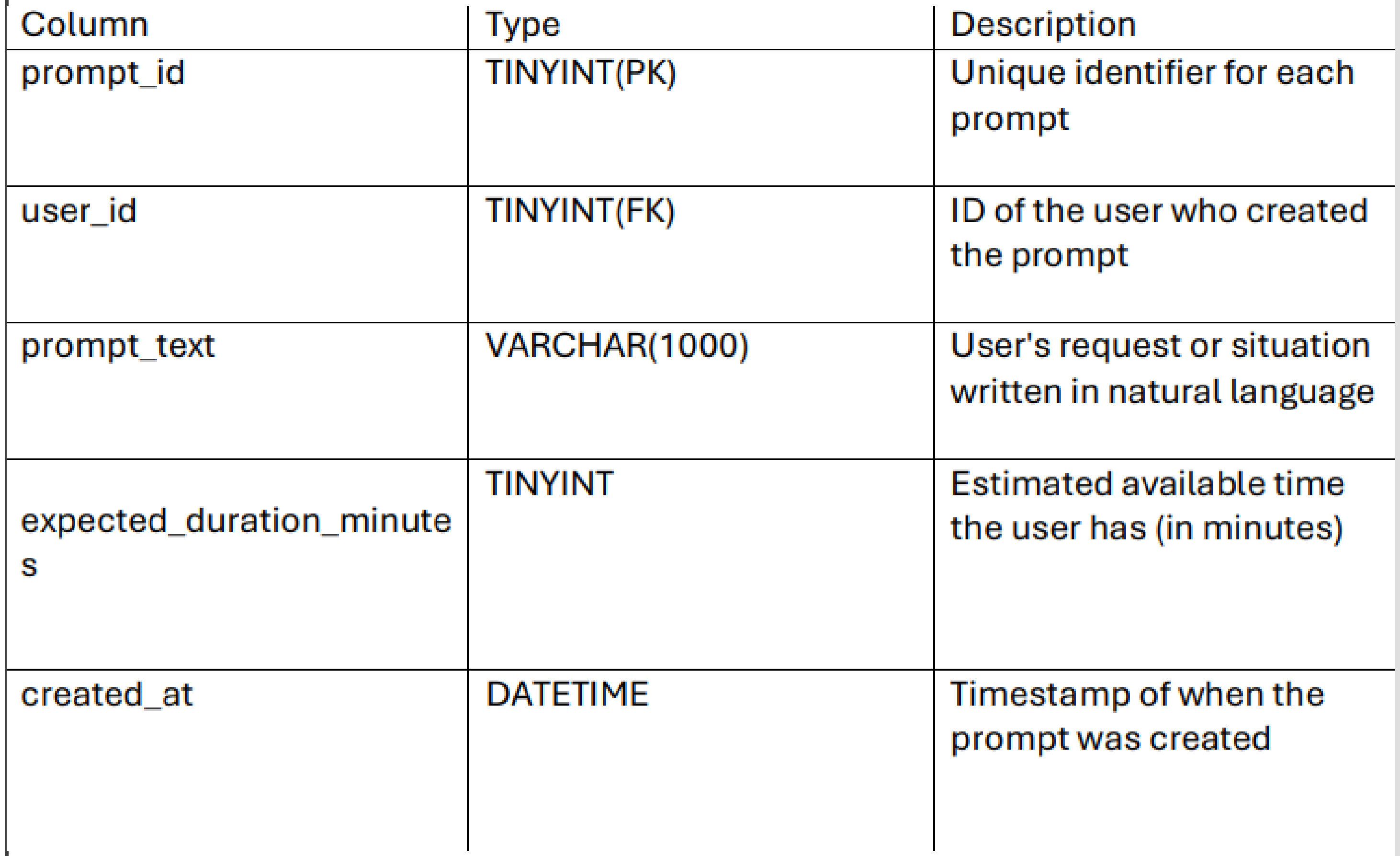
Column	Type	Description
analysis_id	TINYINT	Primary key, unique analysis identifier
user_id	TINYINT	Foreign key referencing Users_1(user_id)
total_free_time	DECIMAL(5,2)	Total free time in hours
total_daily_time	DECIMAL(5,2)	Total daily time in hours
total_busy_time	DECIMAL(5,2)	Total busy time in hours
efficiency	DECIMAL(5,2)	Efficiency percentage

Recommendation Table



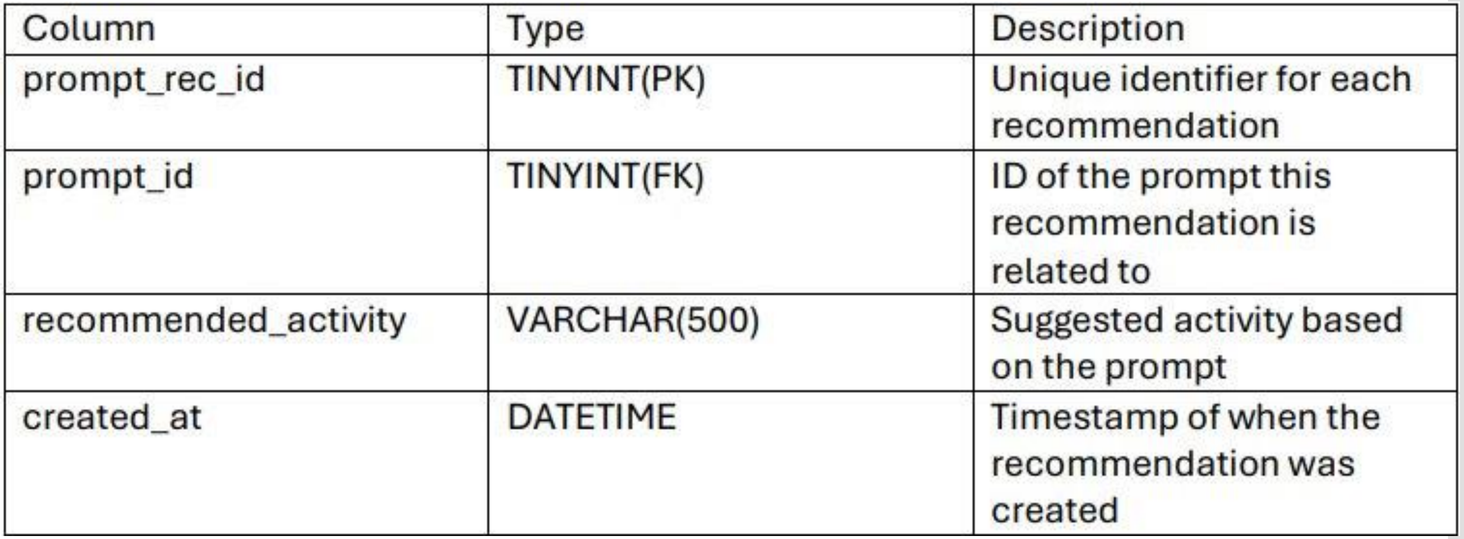
Column	Type	Description
recommendation_id	TINYINT	Primary key, unique recommendation identifier
user_id	TINYINT	Foreign key referencing Users_1(user_id)
activity_id	TINYINT	Foreign key referencing DailyTimeLossActivities(activity_id)
recommendations	VARCHAR(500)	Recommendations provided to the user
created_time	DATETIME	Timestamp when the recommendation was created

User Prompt Table

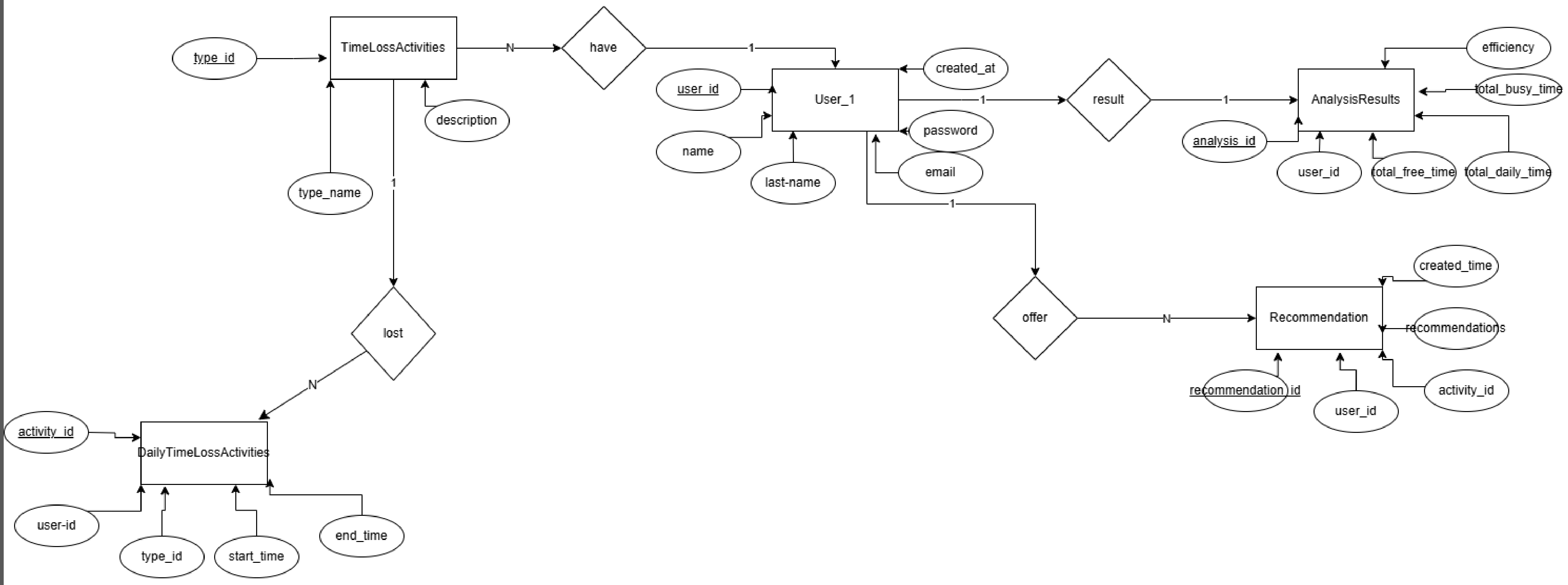


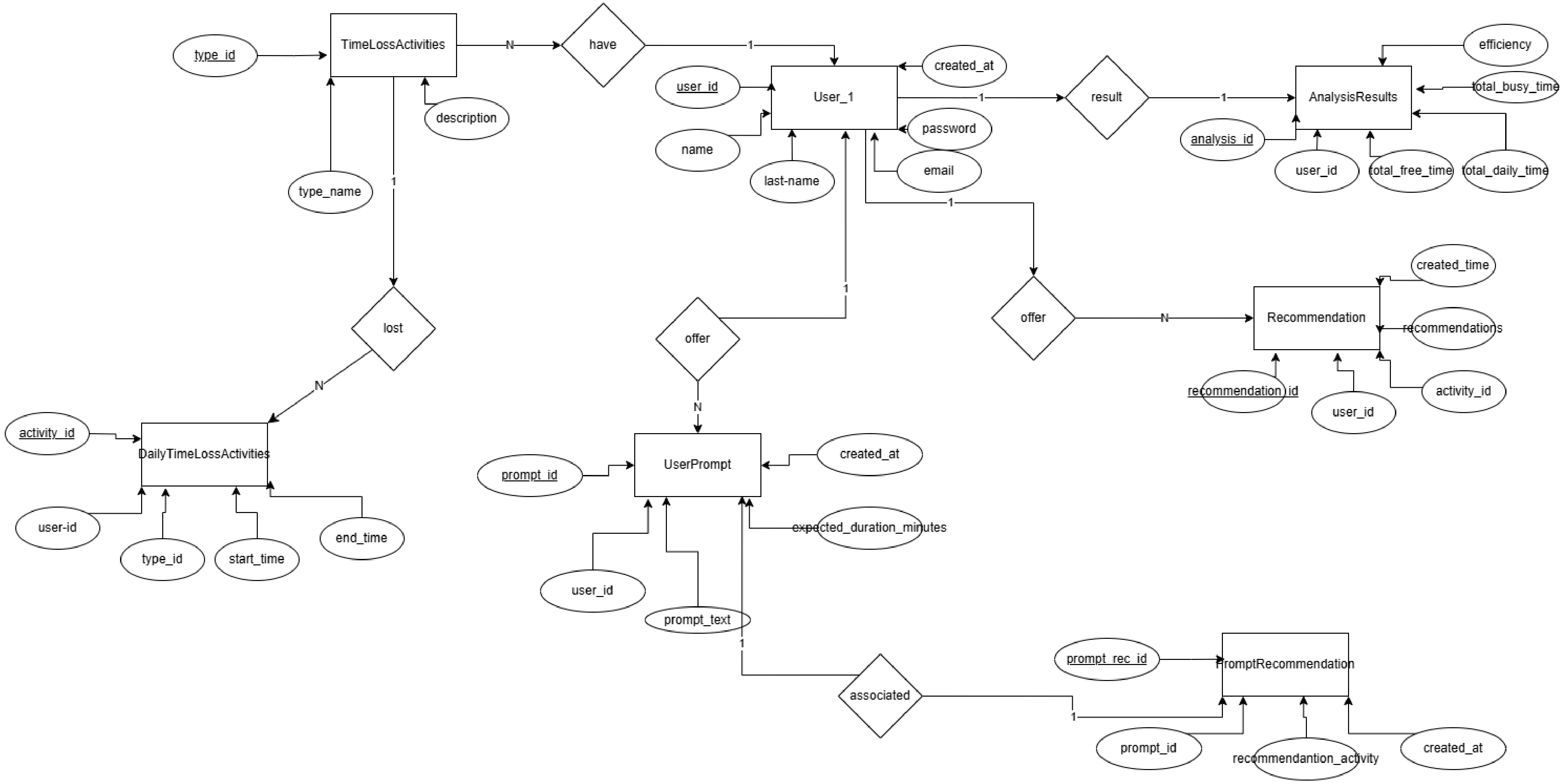
Column	Type	Description
prompt_id	TINYINT(PK)	Unique identifier for each prompt
user_id	TINYINT(FK)	ID of the user who created the prompt
prompt_text	VARCHAR(1000)	User's request or situation written in natural language
expected_duration_minutes	TINYINT	Estimated available time the user has (in minutes)
created_at	DATETIME	Timestamp of when the prompt was created

Prompt Recommendation Table



Column	Type	Description
prompt_rec_id	TINYINT(PK)	Unique identifier for each recommendation
prompt_id	TINYINT(FK)	ID of the prompt this recommendation is related to
recommended_activity	VARCHAR(500)	Suggested activity based on the prompt
created_at	DATETIME	Timestamp of when the recommendation was created


TimeLossActivities
type_id
type_name
description
have
N
1
User_1
user_id
name
last-name
created_at
password
email
result
1
1
AnalysisResults
efficiency
total_busy_time
analysis_id
user_id
total_free_time
total_daily_time
1
offer
N
Recommendation
created_time
recommendations
recommendation_id
user_id
activity_id
1
lost
N
DailyTimeLossActivities
activity_id
user-id
type_id
start_time
end_time


type_id
TimeLossActivities
N
have
1
User_1
created_at
user_id
name
last-name
password
email
description
type_name
1
result
1
AnalysisResults
efficiency
total_busy_time
analysis_id
user_id
total_free_time
total_daily_time
1
offer
N
Recommendation
created_time
recommendations
recommendation_id
user_id
activity_id
lost
N
activity_id
DailyTimeLossActivities
user-id
type_id
start_time
end_time
1
offer
N
UserPrompt
prompt_id
created_at
expected_duration_minutes
user_id
prompt_text
1
associated
1
PromptRecommendation
prompt_rec_id
prompt_id
recommendantion_activity
created_at